DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

Month	Day	Year
01	16	2009

DTIC® has determined on 01 16 2009 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited.

☐ **© COPYRIGHTED.** U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only. Other requests for this document shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors. Other requests for this document shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only. Other requests shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only. Other requests shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by controlling office or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED: *(From – To)*
24-07-2008	AFOSR Final Report	04/01/07 -- 06/30/08

4. TITLE AND SUBTITLE
Model Justified Search Algorithms for Scheduling Under Uncertainty

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-07-1-0403

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Adele Howe
L. Darrell Whitley

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Colorado State University
Fort Collins, CO 80523

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Optimization and Discrete Mathematics
Donald Hearns, Program Manager

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

AFRL-SR-AR-TR-08-0527

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We have identified clear trade-offs in algorithm design for the scheduling under uncertainty problem that relate to the sources of the uncertainty. We also indentified plateaus as a significant barrier to superb performance of local search on scheduling and have studied several canonical discrete optimization problems to discover and model the nature of plateaus. From this, we have developed lower and upper bound predictive models of plateau size in a significant optimization problem: MAXSAT. We also develop new theoretical results on the nature of plateaus in Elementary Landscapes.

15. SUBJECT TERMS
Optimization, Scheduling, Elementary Landscapes

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	C. THIS PAGE			20b. TELEPHONE NUMBER (Include area code)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI-Std Z39-18

Final Report for AFOSR #FA9550-07-1-0403
Model Justified Search Algorithms for Scheduling Under Uncertainty
April 1, 2007 to June 30, 2008

Adele Howe   L. Darrell Whitley
Computer Science Department
Colorado State University
Fort Collins, CO 80524
email: {howe,whitley}@cs.colostate.edu

July 25, 2008

Abstract

Most search algorithms do not explicitly and dynamically exploit problem structure. In fact, most search algorithms throw away all or most of the information that has been collected about the search space. Some algorithms maintain limited persistent state information, but none of these methods explicitly model and exploit problem structure. This means that most of what could be learned about a search space is never exploited.

As part of our long-term research, we have pioneered development of models of the dynamics of scheduling search spaces and used those models to justify the design of new simple search algorithms. In this project, we have focused on thoroughly analyzing an Air Force application of scheduling under uncertainty and have developed new models of a critical feature of discrete optimization search spaces: plateaus. We have identified clear trade-offs in algorithm design for the scheduling under uncertainty problem that relate to the sources of the uncertainty. We also identified plateaus as a significant barrier to superb performance of local search on scheduling and have studied several canonical discrete optimization problems to discover and model the nature of plateaus. From this, we have developed lower and upper bound predictive models of plateau size in a significant optimization problem: MAXSAT. We also develop new theoretical results on the nature of plateaus in Elementary Landscapes.

Contents

1 Project Objectives 3

2 Accomplishments/New Findings 3
2.1 Scheduling Under Uncertainty . . . 3
2.1.1 Algorithms for Spacetrack . . . 4
2.1.2 Assessing the Trade-offs . . . 5
2.2 Modeling Search . . . 7
2.2.1 Elementary Landscapes . . . 7
2.2.2 Plateaus . . . 10
2.2.3 Learning Algorithm Models . . . 12
2.3 New Focused No Free Lunch Theorems . . . 13

3 Executive Summary 15
3.1 Personnel . . . 15
3.2 Publications . . . 15
3.3 Interactions/Transitions . . . 16
3.3.1 Presentations at Meetings . . . 16
3.4 Honors/Awards/Significant Service . . . 16
3.5 Web Site . . . 17

1 Project Objectives

Our long term research goal is to develop the theoretical and empirical underpinnings of a science of automated scheduling. We had two primary objectives for this research project:

1. Analyzing an Air Force relevant application of scheduling under uncertainty and constructing new algorithms that address its sources of uncertainty as well as its novel challenges and constraints.

2. Developing new predictive and explanatory models of search spaces.

To achieve the first objective, we studied an Air Force application: scheduling tracking tasks on the Eglin AFB phased array radar system that is part of the USAF SpaceTrack system. The application imposes significant resource constraints, is over-subscribed, has both a short term and long term objective, and has uncertainty arising from possible task failure. The application supported grounding our research in practical issues and pushed the envelope of research because few studies have examined uncertainty due to task failure. We constructed a problem generator and used it to drive studies into trade-offs inherent in the application. Our new algorithms addressed these trade-offs as well as considering the impact of available time on decision making.

To achieve the second objective, we studied several well known canonical problems: flow shop scheduling, planning and maximum satisfiability. The canonical problems expedited experimental control and development of new theory underlying search. They also supported generalization of research results. We looked at three issues. First, could the theory of elementary landscapes be extended and exploited? Second, because our objects of study are discrete optimization problems, could we model an attribute common to such problems as well as problematic for local search: plateaus? Third, could performance of particular complex search algorithms be predicted from problem features? Our studies identified key features of the problems and characterized them empirically and where possible analytically.

2 Accomplishments/New Findings

This section organizes the major results of our project into two categories: scheduling under uncertainty and modeling search spaces. The first summarizes our results with the SpaceTrack scheduling under uncertainty problem. The second covers the experiments and theory developed from our analyzes of well known canonical problems.

2.1 Scheduling Under Uncertainty

Real world scheduling applications often must be solved with incomplete or uncertain information. We study a real world scheduling application in which orbit tracking tasks must be assigned execution times on a phased array radar at Eglin Air Force Base in Florida, USA. Tracking tasks may fail when they are executed due to uncertainty inherent in the orbital dynamics of tractable entities. The objective then is to maximize a priority weighted sum of tasks that execute successfully. Since the success of a task is not known at schedule generation time, we derive an expectation value that models the task's *a priori* expected contribution to the weighted sum. An object's radar cross section along with its position with respect to the radar's boresight influence detectability. Therefore, the expectation value changes dynamically in time as an object moves through space. Therefore, an individual task's expectation depends directly on the time at which it is scheduled to execute.

We classify scheduling methods for our application into two approaches. In an *on-peak* approach, tasks are assigned the time that maximizes the expectation value during a particular object's pass. In a *relaxed* approach, tasks can be scheduled at any time an object is tractable. Current algorithms for scheduling tracking missions employ an "on-peak" approach. That is, objects are only tracked during the time that produces the optimal probability of success. If the available resource has infinite capacity, on-peak produces the optimal solution.

However, when the resource is sufficiently over-constrained, we prove that the problem becomes NP-hard. In particular, on-peak scheduling can be characterized as an integer programming problem in which all of the selected tasks can be feasibly scheduled on their peaks. This is an instance of the NP-hard $\{0, 1\}$ multidimensional knapsack problem [6]. It contains the traditional (single-dimensional) $\{0, 1\}$ knapsack problem as a special case when the constraint matrix is $1 \times n$. We showed NP-hardness of on-peak scheduling by a reduction from single-dimensional knapsack. Relaxed scheduling is NP-hard since it contains on-peak scheduling as a special case.

The problem constraints also precipitate an interesting interaction. Earlier work [16] suggested that the *on-peak* approach is relatively inflexible and results in more resource contention than the *relaxed* approach. By allowing tracking tasks to occur during times that give potentially suboptimal expectation values, we can empirically increase the overall *utilization* of the schedule. We hypothesize that this expectation/utilization trade-off could be exploited by heuristic search algorithms to produce mission schedules that have a higher *global* expectation value than the optimal on-peak schedule. In such a solution, a subset of individual tasks would have a reduced individual probability of success, but could result in a higher total successful yield at execution time. In general, this cannot always occur; the result depends mainly on the degree of contention and expectation value functions specific to the domain. Thus identifying this property reveals a certain exploitable structure inherent in the problem.

2.1.1 Algorithms for Spacetrack

The intractability of the SpaceTrack domain favors the application of incomplete algorithms. Local search has been effectively applied to NP-hard scheduling problems in which there is an implicit underlying structure that can be exploited by applying a sequence of moves [19,12,18,1]. Our two algorithms are variants of local search that differ in solution construction behavior.

On-peak Local Search finds a subset of on-peak tasks that maximize the evaluation function: sum of the probabilities of success for scheduled tasks estimated from models of the tasks over time. The local search starts with an initial permutation and iteratively applies the *swap* move operator. The $O(n^2)$ size of the neighborhood induced by the swap move operator becomes unmanageable for local search in large realistic instances; we use a *randomized neighborhood.* The permutation is passed to a deterministic schedule builder [21,17,3,11,9,2] which attempts to schedule each task in the permutation on its peak time slot. If a task cannot be feasibly inserted into the schedule, it is discarded, and the next element in the permutation ($\omega(i+1)$) is considered.

Two-phase Local Search further adapts the use a schedule builder to make decisions about 1) which task should be placed in the schedule next and 2) rather than placing the task "on-peak" a secondary search is done to place the task so as to minimize conflicts with other tasks and to maximize the overall yield. To do this, two-phase local search (TP-LS) focuses on two aspects of a candidate solution: an *insertion ordering* for the schedule builder, and an *insertion policy* for each task. The search algorithm is named for its two phase operation. In the *ordering* phase, the algorithm searches the space of schedule builder insertion ordering permutations, in the same way as does the on-peak local search defined above. In the *policy* phase, the algorithm searches for an insertion policy for each task in the schedule builder. The algorithm strategically switches between using search to decide what task to place next, and using search to decide how best to place each task.

When the algorithm is operating in phase 1, it behaves much like OP-LS above, employing stochastic hillclimbing on the insertion ordering on the space induced by the swap operator (although using the policy for the evaluation). When the algorithm switches to phase 2, it enters a *policy mutation* mode in which stochastic hillclimbing is imposed on the placement on the current task being considered for placement in the schedule.

2.1.2 Assessing the Trade-offs

To assess the trade-off and test whether our hypotheses hold in practice, we designed a factorial experiment in which we compare the algorithms on different problem sets and resource capacity constraints. We assess performance or SpaceTrack schedules according to an evaluation function (the Expected Weighted Sum Successful, denoted $\mathbb{E}[WSS]$, which measures the expected "yield" of successful tracks) as well as an objective function which calculates the actual number of successful tasks derived through simulation.

A schedule day for a SpaceTrack radar consists of thousands of objects. We created two problem sets on which to test our hypotheses: an experimental control set and an application consistent set.

The experimental control set provides a collection of small tractable instances on which it is possible to obtain an exact solution to the on-peak schedule in order to make the fairest possible assessment of potential. This set consists of 50 problems of 50 requests each.

We applied to the experimental control set a mixed-integer programming solver (`lp_solve`, which was originally written by Michel Berkelaar at Eindoven University of Technology) that uses LP-relaxation and branch-and-bound. This solver was able to find the on-peak optimal solution for each problem in the experimental control set. This solver could not be applied to the "relaxed" problem which allowed for scheduling off-peak. For the relaxed problem we used the two-phase local search. This set up an interesting question: which approach is better, 1) using an optimal LP solver on the on-peak problem, or 2) using a heuristic method on the "relaxed" problem when optimal solutions cannot be guaranteed?

We found that two-phase search always obtains a mean value higher yield (with the exception of `ec17`) than the optimal on-peak solution found by the LP solver. On average, the mean two phase local search (TP-LS) solutions were 3.6% above OP-OPT (the on-peak optimal solution found by branch-and-bound). Given the small standard deviations observed, the OP-OPT solutions were on average 2.7 standard deviations below the TP-LS mean.

The application consistent set is made up of a set of highly-realistic test problems modeled on real data. We obtained actual data on approximately half of the objects in space that are tracked by SpaceTrack. Our test problems were created using orbital information from NORAD (`http://www.space-track.org`). We generated twelve different problem instances with data from two days in April 2006. The application consistent set is much larger and more realistic and thus cannot be solved by the MIP solver.

The application consistent set is embedded in a simulator. At each scheduling period, we ran each algorithm for 50000 evaluations. The best schedules found by each algorithm were then executed on the simulator which attempts to execute each tracking task; whether or not the task succeeds is proportional to its estimated probability of success.

To simulate auxiliary surveillance and calibration tasks during the course of the schedule day, we computed a set of disjoint intervals such that each interval had a minimum duration of 10 minutes and a maximum of 45 minutes. The actual durations were computed randomly at problem generation time. Each problem instance has about 30 such intervals.

We compared performance of the two phase algorithm to the on-peak algorithm for the application consistent set. To see how each algorithm performs in each scheduling period for a representative instance, we display the (normalized) expected weighted sum successful found by each algorithm for instance `AC5000a` in Figure 1. In each scheduling period, TP-LS performs superior to OP-LS, suggesting that the trend found in the experimental control set translates to the application consistent set.

The improvement produced by the two-phase algorithm was analogous across all instances. In order to summarize the effect, we report the normalized expected weighted sum successful obtained by each algorithm for each problem aggregated over all scheduling periods in Figure 1. Note that the overlap in ranges results from the individual variance across scheduling periods. The gain in $\mathbb{E}[WSS]$ of TP-LS over OP-LS averaged over all sets and scheduling periods for the application consistent set was 30.68% with a standard deviation of 16.06%.

The previous results were computed using the evaluation function which yield an overall expected yield, but does not indicate if a particular task successes or fails. We also looked at performance on the simulated objective function; this this case, each individual task either successes or fails. We performed one-sided Wilcoxon signed rank tests to see whether TP-LS outperforms OP-LS with statistical significance. We compared the mean weighted sum successful values (WSS) and found that TP-LS, using the relaxed scheduling approach, consistently obtained a statistically higher mean weighted sum successful and yield across all instances at the 99.99% confidence level ($p < 0.00001$).

We hypothesized that relaxed insertion offers an advantage when there exists a significant amount of resource contention. As capacity decreases, contention will increase. Figure 2 demonstrates the change in the mean percentage of optimal on-peak solution by TP-LS and OP-LS

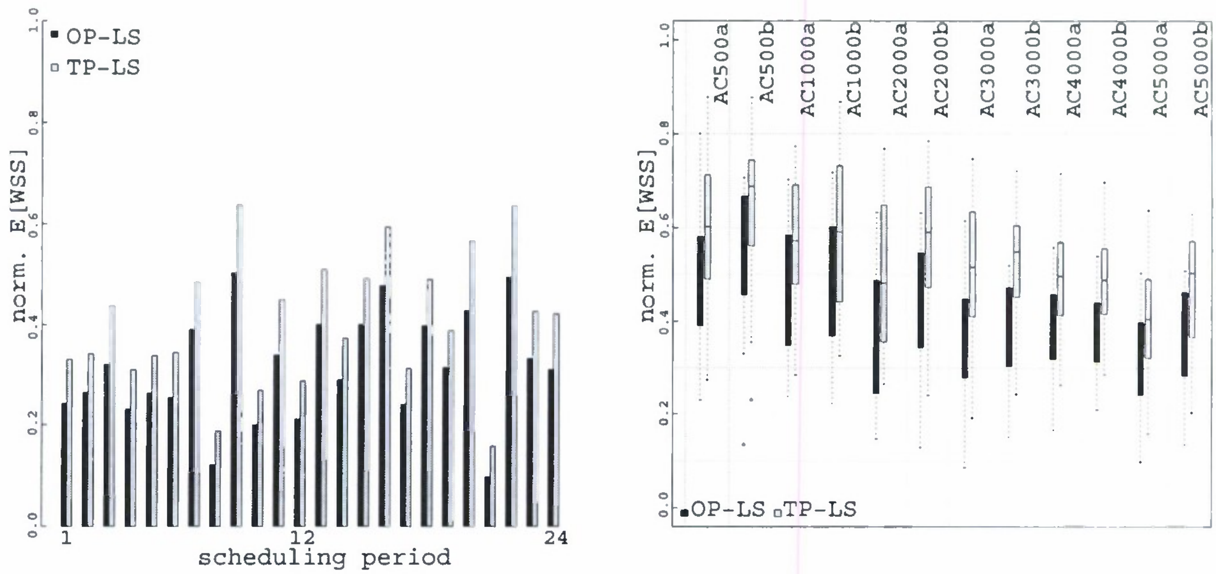

Figure 1: [Left] Normalized $\mathbb{E}[WSS]$ (Expected Weighted Sum Successful) of schedules found by each algorithm in each reschedule period for problem instance AC5000a. [Right] Box and whisker plot of normalized $\mathbb{E}[WSS]$ of schedules found by each algorithm on each instance in the application consistent set.

with respect to this capacity adjustment over all problems in the experimental control set. Note that as the problems become more capacity constrained, the advantage over the on-peak solution exhibited by TP-LS grows significantly.

2.2 Modeling Search

We examined three issues related to modeling search spaces towards the goal of designing search algorithms. First, could the theory of elementary landscapes be extended and exploited? Second, because our objects of study are discrete optimization problems, could we model an attribute common to such problems, specifically plateaus, which is also known to be problematic for local search? Third, could performance of particular complex search algorithms be predicted from problem features? Our studies identified key features of the problems and characterized them empirically and where possible analytically.

2.2.1 Elementary Landscapes

A combinatorial landscape is a formalism that allows us to carefully analyze certain heuristic search algorithms that solve discrete optimization problems. A combinatorial landscape is defined by a tuple (X, N, f) where X is the set of *candidate solutions*, N is the *neighborhood operator* and f is the *objective function* that measures the cost or value of each solution. The structure (X, N, f) can thus be characterized as a vertex-weighted graph in which the vertices

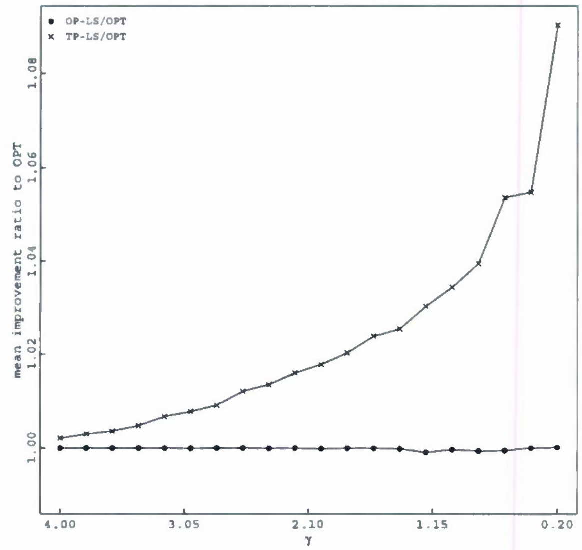

Figure 2: Mean improvement ratio over all capacity controlled problems with respect to capacity adjustment. Capacity decreases from left to right.

are the candidate solutions X weighted by f and connected by the neighborhood operator. A search algorithm can be seen as performing a walk on this graph. Thus analyzing the structure of this graph provides insight into the characteristics of a problem instance described by f and how it might affect different search algorithms.

An *elementary landscape* is a specialization of a landscape that obeys a difference equation. In particular, suppose $a(x)$ gives the *average objective function value* over all the solutions in the neighborhood $N(x)$. Then a landscape is elementary if and only if the following equation holds.

$$a(x) = f(x) + \frac{k}{d}\left(\bar{f} - f(x)\right) \tag{1}$$

where k is constant over the entire landscape, $\bar{f}$ is the average objective function value over X, and $d = |N(x)|$ is the cardinality of the neighborhood (i.e., the degree of the landscape graph).

Equation (1) is important because it induces certain topological constraints on the search space which may ultimately affect how a search algorithm behaves. For example, Codenotti and Margara [4] proved that if Equation (1) holds, for any arbitrary x for which $f(x) \neq \bar{f}$, the entire neighborhood cannot consist entirely of solutions with evaluations equal to f. In other words, there are certain types of plateaus (connected sets of solutions with equal evaluation) that cannot exist on elementary landscapes.

In our research, we have extended this work by discovering two further topological constraints on elementary landscapes. We have also generated new results on MAXSAT modeled as a combination of elementary landscapes.

Result 1: On an elementary landscape, if a solution on a plateau x_1 has only equal and disimproving neighbors, then there cannot exist a solution x_2 with only equal and improving neighbors on the same plateau. Note that $f(x_1) = f(x_2)$ but $a(x_1) \neq a(x_2)$ which contradicts Equation (1) holding for the entire landscape. This also follows directly from a result by Grover [7] which states all local minima lie below the mean landscape evaluation $\bar{f}$ and all local maxima lie above the mean. The implication of this result is that certain forms of plateaus cannot exist on elementary landscapes. For instance, if an algorithm encounters a solution on a plateau with no disimproving moves, we instantly know that *every single solution* on the plateau has an "exit": that is, improving moves to solutions off the plateau. This result holds as long as the landscape is elementary and is not completely flat (i.e., all solutions have the same evaluation) regardless of the size of the plateau.

Result 2: Suppose x_1 and x_2 are two locally minimal solutions on the same plateau. Without loss of generality, we suppose the number of *equal* neighbors of x_1 is greater than or equal to the number of equal neighbors of x_2. Then, assuming minimization, the sum of all disimproving moves in the neighborhood of x_2 must be greater than or equal to the sum of all disimproving moves in the neighborhood of x_1. This follows in a fairly straightforward manner from Equation (1). An analogous result holds for all local maxima on plateaus.

We have also investigated the implications of Equation (1) for search algorithms. Clearly, on elementary landscapes, the average value of the neighborhood $a(x)$ can be computed without actually evaluating any of the neighbors of x. Thus a local search algorithm can assess how promising a solution's neighborhood may be (in terms of expectation) before enumerating the neighborhood. Furthermore, if an algorithm has *partially* expanded a neighborhood, we can immediately compute the average value of the neighbors that have not yet been explored. A perhaps surprising result is that if the average value of a partial neighborhood $P_1 \subset N(x)$ of x is *worse* than the average value of a partial neighborhood P_2, the average value of the remaining neighbors in P_1 is guaranteed to be *better*.

Result 3: Any arbitrary combinatorial landscape can be represented as a linear combination of elementary landscapes. In the worst case, an exponential number of such elementary components are needed (e.g., for a completely random search space). However, it is often the case that many non-elementary landscapes can be represented as a superposition of a relatively small number of elementary components. We have recently proven that MAX-k-SAT can be represented by at most k elementary landscapes. This means the classic MAX-3SAT (often just called MAXSAT) is a composition of only 3 elementary landscapes. We also have conjectured that many other applications that are relatively simple to characterize but difficult to solve are also the combination of a small number of elementary landscapes.

The impact of our research is twofold. In the first place, the guaranteed relationships and forbidden structures in the search space can serve to better inform heuristic search algorithms on how to respond to different problems. This is due to the fact that on an elementary landscape, an algorithm can infer the existence of certain topological relationships without spending computational resources expanding the neighborhood. Second, these analyzes may translate to non-elementary domains by observing how relationships and properties in elementary *compo-*

nents influence the relationships and properties of their superposition. In this case, we expect results about elementary landscapes will indirectly generalize to a large class of important combinatorial optimization problem.

2.2.2 Plateaus

Local search algorithms have been successfully applied to a broad range of problems, including many of the scheduling problems we have studied. In earlier research, we identified plateaus as a dominant feature in the AFSCN scheduling problem; evidence from our studies and others indicate the same is true in other applications as well.

The two characteristics that determine the hardness of escaping a plateau are its *exit density*: the number of strictly improving moves incident to plateau solutions, and its *size*: the number of solutions belonging to the plateau. Since the progress of a local search algorithm is ultimately connected to how well it can escape plateaus, plateau characteristics are intimately related to problem difficulty for local search.

To further understanding of how plateaus influence local search, we developed methods for estimating upper and lower bounds on the expected plateau size for a well defined class of optimization problems: maximum satisfiability (MAXSAT). Such bounds can benefit search algorithms in two ways: first by providing an estimate of how hard a problem instance is likely to be for local search, and second by predicting when the expected size of a plateau is likely to be too large to systematically search.

Under some simplifying assumptions on the distribution of equal valued solutions in the search space, we construct a correspondence between *plateaus* in MAXSAT problems and *percolation clusters* in hypercube graphs. A local search algorithm defines some computationally tractable neighborhood function $N : X \mapsto 2^X$ and, starting from an independently generated initial candidate solution, walks along the graph induced by the neighborhood function. Thus, the behavior of local search can be characterized as a biased walk on the neighborhood graph $G(X, E)$ induced by N, that is, $(x, y) \in E \iff y \in N(x)$. For MAXSAT problems, $G(X, E)$ is isomorphic to a hypercube graph of order n.

A plateau is simply a connected component of the subgraph of the neighborhood graph G induced by a level set. A level set is the maximal set of solutions that have the same value of the objective function.

Lower Bound To derive a lower bound on plateau size, we estimate the size of all neutral paths from an arbitrary point on the plateau. A neutral path is defined as a path through the graph in which all points belong to the same level and that the Hamming distance from the start point is monotonically increasing by unit length at each step in the path. Starting from this, we derived the expected size of the Hamming path set (and therefore our lower bound on plateau size) as

$$\mathbb{E}[|H_x|] = \sum_{r=0}^{n} \binom{n}{r} h_x(r) \tag{2}$$

We develop an estimate of $h_x(r)$ (and so $\mathbb{E}[|H_x|]$) using a percolation approach. Let C_n be a hypercube graph of order n. Each vertex in C_n corresponds to a string $\{0, 1\}^n$. Let

$x = (000...0)$ and $y = (111...1)$. We refer to x and y as the *corner* vertices. A vertex is *active* if it belongs to the same level set as x. We define the *concentration* as the probability p that a vertex is on the same level set as x, and thus active. We assume this probability is constant and independent across all vertices. Note that x is a *fixed active* vertex since it trivially belongs to its own level set. We say the cube *percolates from y to x* if there is a monotonic path $(y = x_1, x_2, \ldots, x_k = x)$ such that all x_i are active.

Let $c(n, p)$ denote the probability that C_n percolates with concentration p from y to the fixed active vertex x. With further analysis,

$$\mathbb{E}[|H_x|] \geq \sum_{r=0}^{n} \binom{n}{r} \hat{c}(r, p) \tag{3}$$

Upper Bound We have characterized plateaus as connected clusters of active sites in the hypercube graph. In this section we will use an exact result from percolation theory to derive an upper limit on the expectation of plateau size for certain values of p.

The Bethe lattice (or Cayley tree) of coordination number n is defined as a connected acyclic graph in which each vertex is connected to n neighboring vertices. For a given concentration p, the expected size of connected clusters of active sites in the Bethe lattice will always be greater than or equal to the expected size of clusters of active sites in the hypercube graph.

The expected cluster size on the Bethe lattice has an exact solution. Let b be an arbitrary active site; the expected cluster size at arbitrary b is:

$$\frac{1+p}{1-(n-1)p} \tag{4}$$

Estimating the Concentration If we know p for a particular level set, then we can bound the expected plateau size. We developed a method that uses a *neutral walk*: a polynomial time algorithm that locally samples around a solution. To estimate p for a level set L, we compute the empirical mean neutral walk length $\mathcal{L}_\mu$ by performing a number of neutral walks from sampled points on L.

Model Validation We have proved that, given our assumptions, our models provide upper and lower bounds. We also empirically assessed how good were these models. To test the size prediction bounds given known concentrations, we evaluate predictions for random hypercube landscapes on which we explicitly control concentration. We report our prediction data in the form of correlation plots. There are three types of data points. "Plateau/HP" is actual plateau size vs. Hamming path prediction. "HP/HP" denotes actual Hamming path set size vs. Hamming path prediction. "Plateau/Bethe" denotes actual plateau size vs. Bethe prediction. A perfect prediction would lie on the diagonal line included in the plots. Data for a 20 dimensional random landscape are plotted in Figure 3. As plateau size increases, we see first increasing and then decreasing inaccuracy for the lower bounds. This is due to our ignoring certain types of paths in our model, which is the subject of future work.

To determine the accuracy of our concentration estimate, we run the above experiments again and estimate p using the neutral walk method. We find a tight correlation between the predicted and actual concentrations.

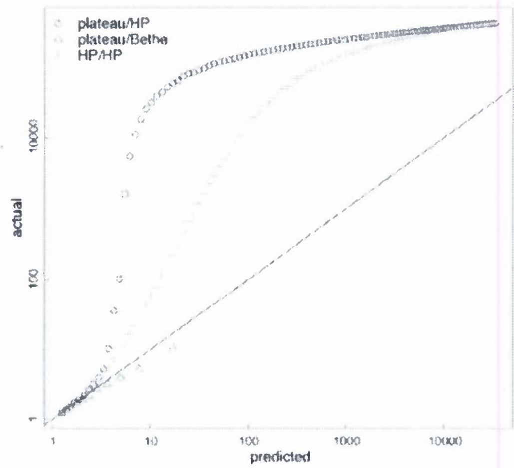

Figure 3: Double log plot of predictions on 20 dimensional random landscape.

To test how well the bounds transfer to actual problems, we perform experiments on random and structured MAXSAT problems. The estimated concentration on each problem set of a particular size appears to decrease as a function of evaluation, which corresponds to observations of other researchers (e.g., [8,5,14].) We also see a marked increase in variance as level increases which suggests plateau size becomes less uniform in better regions of the search space. We found similar trends in accuracy with size across the random and structured problems. Furthermore, the trend is again similar to what we found on the hypercube graph model.

2.2.3 Learning Algorithm Models

Our goal is to design algorithms that are well suited to their applications. We have found in our studies over the years that certain search space features exert a strong influence on the performance of scheduling algorithms (e.g., plateaus) and that knowledge of features can direct the design of new algorithms (e.g., our earlier design of ALLS for scheduling of satellite communications and IJAR for job shop scheduling). In this study, we sought to determine whether models could be learned that predict success of different algorithms on specific problems. Towards that end, we selected planning algorithms as our focus because a large number of systems are publicly available and the community has produced a large number of benchmark problems to support the study.

We collected performance data on 27 planning systems solving almost 5000 problems. No planner solved all the problems. We also collected features of the problems that either could be automatically extracted from their descriptions (e.g., different size metrics and types of constraints) or from partial expansions of their search spaces (e.g., some plateau metrics). Using off-the-shelf machine learning tools, we constructed models for each planner of the likelihood of success and amount of time required to solve problems based on their features. We found that we could create very accurate models of success, but not so accurate models of time. Our post-hoc analysis suggests that time predictions require considerably more knowledge of the dynamics of the search space it interaction with the algorithms than are encompassed by our current feature set.

We also looked at extending existing models of planning search spaces to more general categories of problems and planners. We found that the data validate previous results that link search topology (specifically the incidence of benches and local optima in certain classes of problems) with planner performance on a wider set of problems that those studied earlier.

These results are promising for transfer into scheduling applications and the range of algorithms developed for them. Clearly, the key is to identify predictive features, such as plateau metrics, and obtain data over a wide range of instances.

2.3 New Focused No Free Lunch Theorems

We have developed new proofs [20] which show that a subset of algorithms can have identical performance over a subset of functions, even when the subset of functions is not closed under permutation. We refer to these as *focused sets* and we refer to the proofs under the name *Focused No Free Lunch Theorems.*

Focused No Free Lunch theorems build on the Sharpened No Free Lunch theorem of Schumacher, Vose and Whitley [13] which shows that No Free Lunch holds over sets that are closed under permutation. The No Free Lunch theorem states that no search algorithm is better than another when compared over all discrete functions. The Sharpened No Free Lunch theorem proves that no search algorithm is better than another when compared on any finite set closed under permutation.

The Sharpened No Free Lunch Theorem also proves that any two arbitrarily chosen algorithms will have identical behaviors *if and only if* the set of functions used in the comparisons are closed under permutation.

Unlike the Sharpened No Free Lunch theorem, Focused No Free Lunch theorems hold over sets that are a subset of the permutation closure. At first this seems like a contradiction. But the difference is both subtle and extremely important.

Sharpened No Free Lunch hold when comparing any and all search algorithms. However, if one selects *two specific algorithms* then their behavior may be identical over a much smaller focused set of functions; in some cases the focused set contains only 2 functions and both can be proven to be compressible.

In some cases, a closure exists which corresponds to the orbit of a permutation group. In this case, we leverage mathematical concepts from permutation groups to bound the maximal size of the focused closure. In other cases, particularly when search is limited to m steps, there can be many focused sets and we construct a focused set heuristically.

Ultimately, Focused No Free Lunch theorems are concerned with how researchers and practitioners use and compare search algorithms. If algorithm $A1$ is better than algorithm $A2$ on a benchmark β, the Sharpened No Free Lunch theorem tells us that if we compute the permutation closure of β (denoted by $P(\beta)$), then algorithm $A2$ is equally better than $A1$ on the set $(P(\beta) - \beta)$ in the aggregate. The problem is that usually the size of β is small and $(P(\beta) - \beta)$ is enormous. Focused No Free Lunch results show that there can exists a *focused set* denoted by $C(\beta)$ such that when $A1$ is better than $A2$ on β, then $A2$ is better than $A1$ on $(C(\beta) - \beta)$, where $C(\beta)$ can be quite small.

These results also address the concerns raised by Igel and Toussaint [10] and Streeter [15] who show many broad classes of problems (for example, consider ONEMAX, MAXSAT, trap functions, or N-K Landscapes) are not closed under permutation. They argue that No Free Lunch does not apply in such domains, which restricts the value of this theory. Focused No Free Lunch theorems however, can apply in these domains.

These theoretical results have deep implications for how we test and compare search algorithms.

3 Executive Summary

3.1 Personnel

During the grant period, the following personnel were supported at the indicated level:

PIs:		
Adele Howe	1.5	months
L. Darrell Whitley	1.0	months
Research Assistants:		
Mark Roberts	4.5	full-time months (9 half-time)
Andrew Sutton	9.75	full-time months (10.5 half-time, 4.5 full-time)

3.2 Publications

Journals

- A.M. Sutton, A.E. Howe and L.D. Whitley. "Exploiting Expectation Trade-Off in Probabilistic Modeling with Stochastic Local Search", submitted to Special Issue of *Journal of Scheduling.*

- M. Roberts and A.E. Howe. "Learning from Planner Performance," accepted to *Artificial Intelligence.*

Book Chapters

- D. Whitley, A. Sutton, A. Howe and L. Barbulescu. "Resource Scheduling with Permutation Based Representations: Three Applications." In *Evolutionary Computation in Practice*, T. Yu and L. Davis, Eds., Springer Berlin, pp. 219-243, 2008.

Conferences and Workshops

- A. Sutton, A.E. Howe and L.D. Whitley. "Using Adaptive Priority Weighting to Direct Search in Probabilistic Scheduling", in *Proc. of International Conference on Automated Planning and Scheduling 2007*, Providence, RI.

- L.D. Whitley, A.M. Sutton and A.E. Howe. "Understanding Elementary Landscapes", in *Proc. of the Gentic and Evolutionary Computation Conference (GECCO-08)*, Atlanta, GA, July 2008.

- D. Whitley and J.R. Rowe. "Focused No Free Lunch Theorems", in *Proc. of the Gentic and Evolutionary Computation Conference (GECCO-08)*, Atlanta, GA, July 2008.

- M. Roberts, A.E. Howe, M. desJardins and B. Wilson, "What Makes Planners Predictable?", to appear in *Proc. of the International Conference on Artificial Intelligence planning and Scheduling Systems*, Sydney, Australia, September 2008.

- M. Lunacek, L.D. Whitley and A. Sutton. "The Impact of Global Structure on Search", *Proc. of the 10th International Conference on Parallel Problem Solving from Nature.* Dortmund, Germany 2008

- R. Dewri, D. Whitley, I. Ray and I. Ray. "Optimizing Real-Time Ordered-Data Broadcasts in Pervasive Environments Using Evolution Strategy", *Proc. of the 10th International Conference on Parallel Problem Solving from Nature.* Dortmund, Germany 2008

- R. Dewri, I. Ray, I. Ray and D. Whitley. "Security Provisioning in Pervasive Environments Using Multi-objective Optimization", to appear in *European Symposium on Research in Computing and Security*, Malaya, Spain, 2008

3.3 Interactions/Transitions

3.3.1 Presentations at Meetings

A. Sutton Oral presentation of "Differential Evolution and Non-separability: Using selective pressure to focus search", GECCO 2007, London, England, July 11, 2007 ;Oral presentation of "Using Adaptive Priority Weighting to Direct Search in Probabilistic Scheduling", at International Conference on Automated Planning and Scheduling, September 24 2007, Providence, RI; oral presentation of "Analysis of Search Landscape Neutrality in Scheduling Problems" at ICAPS 2007 Doctoral Consortium, Providence, RI, September 22 2007; oral presentation of "Understanding Elementary Landscapes" at GECCO 2008, Atlanta, GA, July 15 2008.

A. Howe and L.D. Whitley "Model Justified Search Algorithms for Scheduling Under Uncertainty: Progress Report" talk at annual AFOSR PI meeting, April 2008 in Arlington, VA.

A. Howe oral presentation entitled "Learned Models of Performance for Many Planners" at ICAPS2007 Workshop on Artificial Intelligence Planning and Learning, Providence, RI, September 2007.

L.D. Whitley invited tutorial entitled "No Free Lunch for Search" and oral presentation of the paper "Focused No Free Lunch Theorems" at GECCO 2008, Atlanta, GA, July 12 2008; invited tutorial, GECCO conference, London, July 11, 2007; invited speaker, "Relating Theory and Experiments", Dagstuhl 2008, Theory of Evolutionary Computation Workshop.

3.4 Honors/Awards/Significant Service

A. Howe Program Co-Chair for Twenty-Second Conference on Artificial Intelligence (AAAI 2007), Associate Editor for *Journal of Artificial Intelligence Research*

L.D. Whitley Chair of the Governing Board for ACM Sigevo. Member of the ACM Sig Governing Board. Co-Chair of a workshop for new CS department heads at the Computing Research Association "Snowbird" conference. Associate Editor for *Theoretical Computer Science,* and *Evolutionary Computation.* Editorial Board, *Journal of Heuristics.* Co-chair and Co-editor of Foundations of Genetic Algorithms, 2007.

3.5 Web Site

Our project web site is available at `http://www.cs.colostate.edu/sched/`. From that site, you can access publications and data from the project.

References

[1] Laura Barbulescu, Adele E. Howe, L. Darrell Whitley, and Mark Roberts. Trading places: How to schedule more in a multi-resource oversubscribed scheduling problem. In *Proceedings of the International Conference on Planning and Scheduling*, 2004.

[2] Laura Barbulescu, Jean-Paul Watson, L. Darrell Whitley, and Adele E. Howe. Scheduling space-ground communications for the air force satellite control network. *J. Scheduling*, 7(1):7–34, 2004.

[3] Christian Bierwirth. A generalized permutation approach to job shop scheduling with genetic algorithms. *OR Spectrum*, 17(2):97–92, 1995.

[4] B. Codenotti and L. Margara. Local properties of some NP-complete problems. Technical Report TR 92-021, International Computer Science Institute, Berkeley, CA, 1992.

[5] Jeremy Frank, Peter Cheeseman, and John Stutz. When gravity fails: Local search topology. *Journal of Artificial Intelligence Research*, 7:249–281, 1997.

[6] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[7] Lov K. Grover. Local search and the local structure of NP-complete problems. *Operations Research Letters*, 12:235–243, 1992.

[8] Steven Hampson and Dennis Kibler. Plateaus and plateau search in boolean satisfiability problems: When to give up searching and start again. *DIMACS Series in Discrete Math and Theoretical Computer Science*, 26:437–453, October 1993.

[9] Emma Hart, Peter Ross, and Jeremy Nelson. Solving a real world problem using an evolving heuristically driven schedule builder. *Evolutionary Computation*, 6:61–80, 1998.

[10] C. Igel and M. Toussaint. On classes of functions for which No Free Lunch results hold. *Information Processing Letters*, 86(6):317–321, 2003.

[11] Soraya Rana, Adele E. Howe, L. Darrell Whitley, and Keith Mathias. Comparing heuristic, evolutionary and local search approaches to scheduling. In B. Drabble, editor, *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, pages 174–181. AAAI Press, 1996.

[12] Mark Roberts, L. Darrell Whitley, Adele E. Howe, and Laura Barbulescu. Random walks and neighborhood bias in oversubscribed scheduling. In *Multidisciplinary International Scheduling: Theory and Applications (MISTA 05)*, pages 98–106, New York, NY, July 2005.

[13] C. Schumacher, M.D. Vose, and L.D. Whitley. The no free lunch and problem description length. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 565–570, 2001.

[14] Kevin R. G. Smyth. Understanding stochastic local search algorithms: An empirical analysis of the relationship between search space structure and algorithm behaviour. Master's thesis, University of British Columbia, 2004.

[15] Matthew Streeter. Two broad classes of functions for which a No Free Lunch result does not hold. In J. D. Schaffer, editor, *GECCO 2003*, pages 1418–1430. Springer LNCS, 2003.

[16] Andrew M. Sutton, Adele E. Howe, and L. Darrell Whitley. Spacetrack: trading-off quality and utilization in oversubscribed schedules. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-2006)*, 2006.

[17] Gilbert Syswerda and Jeff Palmucci. The application of genetic algorithms to resource scheduling. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of Int'l Conf. on Genetic Algorithms (ICGA)*, pages 502–508. Morgan Kaufmann, 1991.

[18] Éric Taillard. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operations Research*, 47:65–74, 1990.

[19] Jean-Paul Watson. *Empirical Modeling and Analysis of Local Search Algorithms for the Job-Shop Scheduling Problem.* PhD thesis, Colorado State University, Department of Computer Science, 2003.

[20] D. Whitley and J. Rowe. Focused no free lunch theorems. In *GECCO-08*. ACM Press, 2008.

[21] L. Darrell Whitley. The GENITOR algorithm and selection pressure. In *Third International Conference on Genetic Algorithms*, San Mateo, CA, 1989. Morgan Kaufman.